How JS works behind the scenes?

How JavaScript Works

JavaScript operates in a single-threaded environment, meaning it executes one task at a time. It handles the execution of code in Execution Contexts (EC), which can be either Global or Functional.

Global Execution Context (GEC)

- Created when the code file is loaded.
- The GEC is responsible for setting up the global object (in browsers, it's window) and the (this) keyword.
- All variables and functions declared in the global scope are stored in the Global EC.

Execution Context Phases

- Memory Creation Phase:
  During the memory phase, JavaScript allocates memory for variables and functions. Variables are assigned the value undefined initially, while functions are stored in memory with their full definition.
- Execution Phase:
  After the memory allocation phase, JavaScript starts executing the code line by line. Variables are assigned actual values and functions are executed.

Functional Execution Context (FEC)

- Whenever a function is invoked, JavaScript creates a new Execution Context specifically for that function.
- Execution context for each function has two phases:
- Memory Phase: Memory is allocated to the local variables, and functions inside the function.
- Execution Phase: The function body is executed, and variables get their actual values.
- After execution, the function's execution context is destroyed, freeing up memory.

Return Statements
Once a function reaches a return statement, the function immediately exits, and its execution context is removed from the Call Stack. Any code written after a return statement won't execute.

The Call Stack
JavaScript uses the Call Stack to manage execution contexts.

- LIFO (Last In, First Out) concept is used:
- When a function is called, its execution context is placed on top of the stack.

- Once the function completes (or reaches return), its execution context is popped off the stack.

JavaScript is Single-Threaded

Single-threaded means JavaScript can only perform one operation at a time but still it can perform asynchronous operations isn't that interesting in my next post we will discuss about this amazing functionality of JavaScript.